A Novel Navigation Scheme in Dynamic Environment Using Layered Costmap

Xiaoning Han[1,2], Yuquan Leng[1,2], Haitao Luo[1], Weijia Zhou[1]

1. State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016

2.University of Chinese Academy of Sciences, Beijing, 100000
E-mail: {hanxiaoning, lengyuquan, luohaitao, zwj}@sia.cn

Abstract: Navigation is one of basic functions of auto-mobile robots. After dozens of years of development, now the navigation in static environment almost has been realized, taking the method in ROS (*i.e.* Robot Operation System) navigation stack as an example. However, when cruising in dynamic environment, there are more difficulties, as the objects in the environment can change their positions. To deal with such cases, we propose a scheme to navigate using layered costmap. By predicting the dynamic object's encounter position according to both its and the robot's kinematic information, and put it in another layer, and set different costs around the objects according the estimation of the motion of them, then path based on the layered costmaps can be planned. In this paper, the safety and efficiency of our scheme have been proven by the results of simulation.

Key Words: Dynamic Navigation; Layered Costmap, Automobile Robots, Path Planning

1. INTRODUCTION

As the one of basic functions of auto-mobile robots, navigation has been a focus of robot engineers and researchers for dozens of years. The navigation in static environment has been already widely used. Taking the ROS navigation stack for example, there are several successful navigation algorithms have been developed on the base of it. And in the recent years, navigation in dynamic environment has drawn many researchers' interests.

Navigation in dynamic environment is much more difficult than in static. While cruising in the dynamic environment, without particular algorithm to deal with dynamic objects, robot would treat them as static ones, which makes it difficult to find an accessible path. For example, when a person walks from point A to point B, it could be viewed as a "wall" formed by the person trajectory, as shown in Fig 1, thus it can be impossible to find a path through the trajectory of the person. However, a person only takes a small part of space at a very moment, and the rest space is available for navigation. It is unreasonable to treat dynamic objects as static ones. Therefore, there is a need for a robot to treat static objects and dynamic objects respectively in its environment.

There is another challenge to express moving objects in maps. As we mentioned above, moving objects change their locations as time goes, and they should be expressed depending on time. Sometimes it is hardly to communicate with other dynamic objects, because of lack of communication equipment or allowance of time, like at the first time encountering an alien robot or person, thus the most valuable information is the observation of the robot. However, the observation is always obtained with noise and

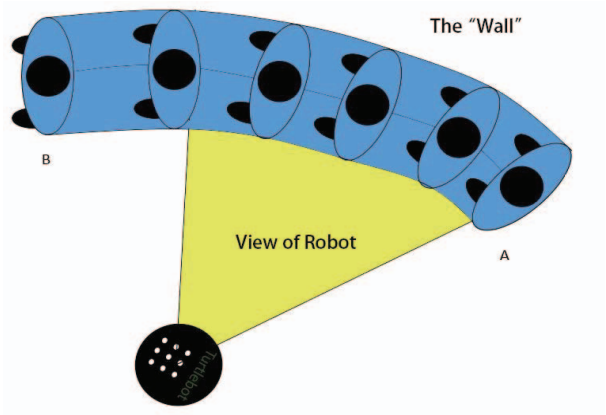

Fig 1 The "wall" formed by person trajectory when treated as a static object

the prediction cannot be perfect. If the motion pattern could be determined upon noisy sensor measurements, the navigation will be significantly simplified.

Actually, the motion of a person or dynamic object can't simply be expressed by mathematical forms, as they has to revise their path according to environment in real time. While in open space, like in hall of hospital, the motion pattern can be simple as moving straight at constant velocity, the kinematic models are applied to predict future motion with only recent motion information is available. With those information, the kinematic parameter can be computed, like location, velocity and acceleration. In more general case, we can only get recent locations of moving objects, which are time series. Thus a kinematic model based recent location series is used in moving objects motion prediction.

This work is supported by National Nature Science Foundation of China under Grant 51505470

Based on the *Encounter time* concept, we proposed a navigation scheme for dynamic environment, which is based on prediction with constant velocity motion model and layered costmap.

The rest of article is organized as following: we review the development of related works, include moving objects detection and tracking, kinematic motion prediction model and layered costmap in section 2. The details of our method are shown in the section 3, in which the concept of encounter time is defined. And we have proven the safety and efficiency of our scheme with simulations in section 4. In the section 5, the advantages and limitations are summarized in the conclusion.

2. RELATED WORKS

The proposed scheme is based on the following 3 technologies: moving objects detection and tracking, motion prediction and layered costmap.

2.1 Moving Objects Detection and Tracking

Moving objects, especially people detection and tracking has drawn many researchers' interests for over 10 years. In early research in the beginning of 21st century, 2-D range data were used to do this task. In [16], Bayesian filtering has been applied to track a number of moving objects with the 2-d perceptual range of robot. Arras began to use learning technique, based on simple features and identifiers, to design classifier, and their experiments in the cluttered office had illustrated robustness of their mothed with an encouraging detection rates over 90% [1].

And then the 3-D range data were preferred, as the data can provide more information. As in [13] and [14], 3-D LADAR scan data have been used to analyze the people. Statistical recognition techniques are proposed with geometric and motion features, which are extracted from three-dimensional LADAR data [14].

In recent years, with the popularity of RGBD sensors like Microsoft Kinect and Asus Xtion, the method using RGBD information has become mainstream. In work [17], combo-HOD is proposed, which is a combination of both HOG (Histogram of Oriented Gradient) and HOD (Histogram of Oriented Gradient), the former is based on RGB images and the latter on depth information, their detector can achieve an Equal Error Rate at 85% and "run at 30 fps on a graphics card implementation". A model based approach, which matches human head with a 2-D head contour model and a 3-D surface model, tried to extract whole human contours from surroundings [19]. With the development of OpenNI framework and OpenCV library, the people could be detected easily [8].

2.2 Kinematic Motion Prediction Models

To analysis the motion of the moving objects, different types of models has been proposed [18, 7, 9]. According to review [6], these methods could be three types: physical-based prediction methods, statistical-based prediction methods and cooperative-based prediction methods. And we will focus on physical-based prediction methods according to robots' work environment assumption.

In the very beginning, the linear function was used:

$$l(t_q) = l_0 + v_0 \cdot (t_q - t_0) \tag{1}$$

where l and v donate location and velocity of moving objects. Given location $l(t_0)$ and velocity v_0, the future location $l(t_q)$ can be predicted. While the objects' motion is complex, the non-linear functions are preferred. There are some works focus on non-linear function to describe more sophisticated motion pattern. As in [15], a polynomial function was used, and the coefficients were computed with a neural networks.

Noisy sensor measurements are important problems for motion models, so there are some works try to deal with uncertainty[4, 2]. To deal with this problem, A. Elnagar employed Kalman filter in motion model[3].

2.3 Layered Costmap

The idea of layered costmap comes from the occupancy grip, which was proposed in 1980s[12]. Its basic thought is that the whole map is separated into numerous cells, each cell contains a cost value which is probabilistic estimate of whether it is empty or occupied by an object in environment. After that, more kind of costs are imported in, especially for soft constraints, whose values are between free and occupied. Those soft constraints have been successfully used in auto-driving[5] and human-robot interaction[10]. The traditional monolithic costmap is successful to deal with some static environment, but struggles to deal with dynamic environments with people.

To overcome the simplicity of monolithic costmap, layered costmaps has been employed to separate the processing of costmap data into semantically-separated layers, with each layer tracking one type of obstacle or constrain. All layers would be integrated by a master costmap which is applied for path planning. And in this work, proxemics layer was described to ensure robot keep the spatial relations with people [11].

3. SCHEME ARCHITECTURE

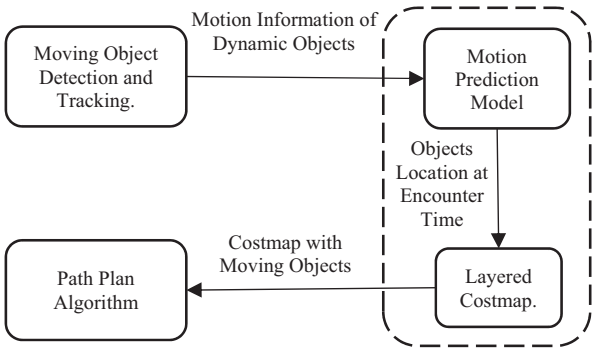

Fig 2 The Architecture of the Dynamic Navigation Scheme

An overview of our scheme is shown in Fig 2, the dynamic navigation scheme consists of four parts: moving objects detection and tracking, motion prediction model, layered costmap and path planning algorithm. As there are suitable solutions for the first and last part, our scheme focus on the prediction model and layered costmap. And we propose a

concept of "*encounter time*" to converse dynamic cases into static ones, and thus the worst situations can be avoided.

3.1 Motion Prediction Model

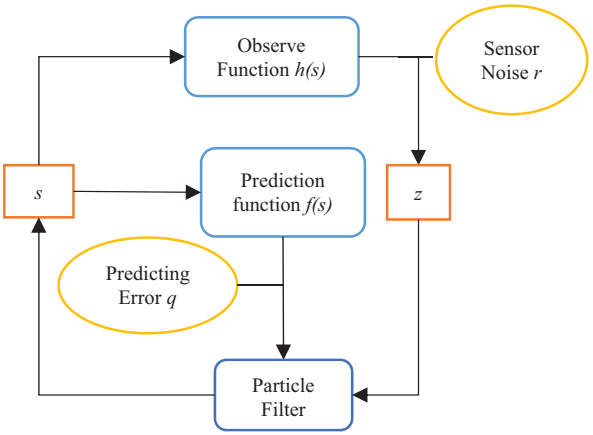

Fig 3 The Motion Prediction Model with Particle Filter Dealing with Sensor Noise r and Predictiong Error q

Motion prediction has great effects on performance of navigation schemes. In short term prediction, constant velocity model and constant acceleration model are widely used, as they can guarantee enough precision. Moreover, the observation of objects is always come with noise and error, and the error will accumulated to be unbounded without correction. Thus we import particle filter into motion prediction model to limit our observation error. Therefore, our motion prediction model includes such two contents: constant velocity model and particle filter.

We assume the robot is running in a wide square hall, so the constant velocity presume is reasonable. The robot state x_k at time t_k can be predicted, as the t_{k-1} state is known.

$$s_k = f(s_{k-1}) + q_k \tag{2}$$

where q_k is the predicting error, and the particular form x for a robot in a 2-D environment is

$$s = \begin{bmatrix} p \\ v \end{bmatrix} \tag{3}$$

and p is 3×1 vector of the robot, $p = [x \quad y \quad \theta]^T$, includes the location in Cartesian coordinates and the orientation, the while v is velocity vector, $[v_x \quad v_y \quad \omega]^T$. the prediction function f is written as

$$f\left(\begin{bmatrix} p \\ v \end{bmatrix}\right) = \begin{bmatrix} s + v \bullet \Delta t \\ v \end{bmatrix} \tag{4}$$

Moreover, the moving objects are observed with the sensing function h, the observation z_k depends on real state s_k, and almost unavoidably, the observations come with sensor noise r_k. Thus the observe function is expressed as

$$z_k = h(s_k) + r_k \tag{5}$$

In general cases, the only information observable is poses, which means $z_k = [x \quad y \quad \theta]^T$ and the form of h is

$$h\left(\begin{bmatrix} p \\ v \end{bmatrix}\right) = p \tag{6}$$

To deal with predicting error q and sensor noise r, we would better to look into them first. If they are Gaussian form, the filter is suitable to deal in Kalman Filter. However, which is more often, they are not Gaussian form and we have little information about their distribution. Thus the particle Filter is preferred to deal with such cases.

3.2 The Encounter Time

As the moving objects march on their trajectories, it is hard for the robot to get its path. To deal with this situation, we propose a novel definition as *encounter time*.

Encounter time for two moving objects is the moment t_e when the two moving objects are closest to each other in their trajectories. And in mathematical form:

$$t_e = \arg\min_t D(s_r(t), s_m(t)) \tag{7}$$

Where $s_r(t)$ and $s_r(t)$ represent the state of the robot and moving object respectively, and $D(*)$ is a function measures the distance between them.

To attain the encounter time t_e, the $s_r(t)$ and $s_m(t)$ should be known. $s_r(t)$ is about the robot trajectory, which is always available or can be estimated easily. While $s_m(t)$ is the state of a moving object, which can predicted with the model we introduced in last subsection.

Then at the encounter time, the robot is closest to moving object, the moment is regarded as the most hazardous moment for the robot, so it is beneficial to know where objects are at the *encounter time* and where the object will be at that time. And based on the *Encounter Time*, we would like to mark the space around the robot location at that time as encounter area, by some inflation algorithm, which could be adjusted in layered costmaps.

3.3 Layered Costmaps

With the observations and initialized motion model, the locations of moving objects can be predicted. And there is a need to be shown in costmap for path planning. Moving object's location depend on time, which is actually a function with a time parameter.

It is almost impossible to deal with the time-based function, as the path planning is based on static cases. And in fact, what matters is that moment when they close to each other.

With the prediction model and recent location series, the robot state s_e at encounter time t_e can be predicted as following:

$$s_e = f(t_e) \tag{8}$$

the location l_e is included in the robot state s_e ..

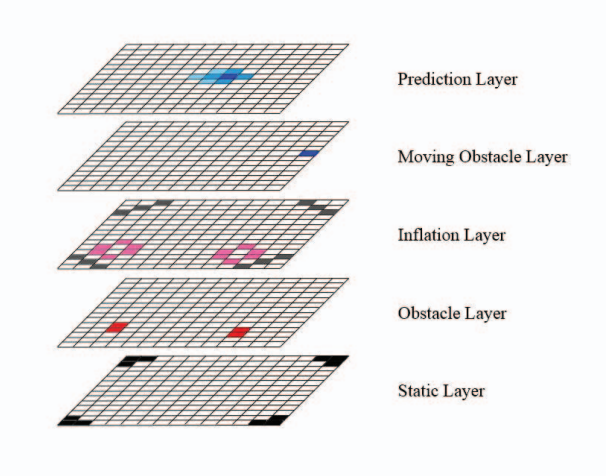

Fig 4 The Costmap System of Multiple Layers.

As shown in Fig 4, there are some conventional layers in the costmap system. The static layer is a layer to show the static environment, which includes walls, and pillars, which is often known before navigation mission. The Obstacle Layer is designed to include static obstacle which is observed during navigation by LADAR or other perceptive sensors. The *Inflation Layer* sets cost value at the cells around those in Static Layer and Obstacle Layer according to the robot's configuration, the design idea of this layer is to guarantee a safe distance between robots and those lethal obstacles. Those traditional layers provide necessary information for a robot to cruise in a static environment.

To import predicted location into map and treat it respectively, we devised two novel layers called *Moving Obstacle Layer* and *Prediction Layer*, Moving Obstacle Layer contains the current location of moving objects, and *Prediction Layer* contains the encounter area. It is beneficial not to pass in front of the moving objects, as they might not have abilities to deal with dynamic objects. Those additional layers improve the navigation scheme safety and robustness.

There are two steps to update the master costmap. In the first step, all layers are look through to get the update range in master costmap with *updateBounds* method, which finally returns a minimum rectangle which contains all cell to update values in all layers. Following the update bound step, the cost will be updated. The cell cost value in update bound will be cleared, and then every layer update its cells value in update bound in order, with the *updateCosts* method. After these two step, the master costmap has been updated with all change in different layers. With the update process runs at a constant frequency, the costmap can be used by robot to plan its path efficiently.

4. SIMULATION

To prove the efficiency of our method, we use ROS to simulate the experiment, and choose the minimal distance and time cost as two evaluation indicators for our scheme. And based on the simulation results, the proposed scheme is analyzed.

4.1 System Setup

The simulation is executed in PC, with Ubuntu 14.04 and ROS Indigo on it. And the global and local path planner is default in ROS.

We use an empty map to simulate open space like hall in hospital. And use *rosbag*, which is message record and play software package provided by ROS, to publish moving objects information message on the topic, where the robot listens to. The information includes the current pose, velocity and unique id for the each moving object. In this way, the detecting and tracking procedure is simulated. More exactly, the map we use is a 10×10 square in ROS stage stack, with the original of world coordinate set on its center. The message is about an object, which is moving from (-4.5, 0.5) along the line $y=0.5$ with a velocity (0.5, 0). While earlier or later, the robot at (3, 3) gets a goal to move to (3, -3). As the robot marches towards the goal, there will be an encounter between the robot and moving object, thus our scheme can be examined by observing how the robot deal with such cases.

4.2 The Indicators and Situation

As the simulation is introduced above, we took 99 experiments and collected basic data. To show the safety and efficiency of our navigation scheme, we choose two indicators to evaluate: 1) the minimal distance between the robot and moving object during their encounter, and 2) the time cost from the robot receiving the mission and the robot reach the goal. The larger distance can guarantee collision-free more robustly, so the minimal distance shows the most hazardous case during the encounter. As the robot almost always has to avoid moving objects actively, it will take more time than when it marches without objects in its way. By measuring the time it takes during each navigation, we can have an idea of how efficiency our scheme is. Therefore, the minimal distance and time cost are chosen as two indicators to measure safe and efficiency respectively.

There are three typical encounter situations in reality when the robot and moving object would get through the encounter area: 1) the robot get there first, 2) both the robot and moving object get there almost simultaneously, and 3) moving object get there first. There are differences among three situations. By setting the robot navigation tasks at different time, the three situations are simulated.

4.3 Simulation Results and Analysis

To examine our dynamic navigation scheme, we have taken 99 simulation experiments. In Table 1, the statistic information of minimal distance and time cost is shown, the average minimal distance is 0.924 and the average time duration is 18.146s. Therefore, our scheme has a generally good performance of safe and efficiency. Though there is a very case that the minimal distance is 0.328, which is very dangerous for the robot and moving object, which happened in an "almost synchronic" case with an obvious "hesitation". There are only 4 such cases that the minimal distance is less than 0.6 in our experiments, which can guarantee safety robustly.

Table 1 Statistic Information of Simulations Results

	Minimal distance	Time cost(s)
average	0.92	18.15
min	0.33	12.00
max	1.55	31.00

As the motion prediction provides estimation of moving objects' future states, we can use this information to foresee the worst situation, i.e. encounter time. By importing the prediction layer the scheme, it is convenient for the robot to treat dynamic objects as static ones. In addition, the social inflation process added can keep the robot away from the encounter area. And as the dynamic objects are treated as static ones, it's convenient for the robot to get path plan quickly with conventional path planning method. Thus the scheme can guarantee both safety and efficiency as our simulation results shows.

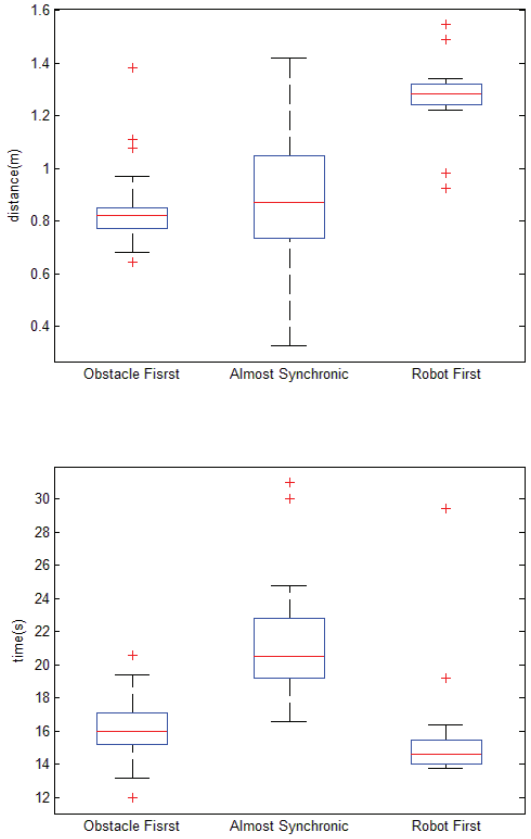

Fig 5 The Simulaiton Results in Three Situations (The upper shows minimal distance information between robot and moving objects and the bottom indicates time cost of navigation mission)

As shown in Fig 5, the robot has significant performance differences in three situations. In the "Almost Synchronic" cases, as the robots always have hesitations during encounter with objects, the performance are worst, and has a little better performance in minimal distance, but a highest average time cost have been taken. While in Robot First cases, where the performances of the robot are pretty good, has an average minimal distance at 0.89 and the average time cost is lowest at 21.14s.

Actually the "hesitation" is not a simply waiting state, but rather some useless attempts to get optimal path. More exactly, the robot is attempting to get optimal static path planning, while it destroys the attempts whenever it changes its motion, thus it runs into an endless cycle until the moving objects pass through the encounter area, and the endless cycle would be broken. After that, the robot will get path plan and navigate as in a static environment. It is understandable that the "Almost Synchronic" cases have almost same average minimal distance as "Object First", but have higher time cost.

Another interesting thing is the "Robot First" cases have better performance than "Object First", especially in time cost. As the prediction layer converts it into static area, the robot should have almost same performance. Actually, the difference is caused by social inflammation, which tries to keep the robot not affect moving objects' motion, by improve the cost in front of moving objects'

5. Conclusion and Future Work

Using the concept of "encounter time", we converse the dynamic objects as virtual static ones, thus the previous static-environment based path-planning can be used. As the encounter time is the most hazardous moment during the navigation, so the idea of the particular moving object layer is to guarantee avoid the worst case. Higher social inflation value can keep the robot more far away for the moving object, so the additional social *Inflation* added can adjust the scheme robustness easily. In simulations, the proposed scheme has reasonable performance, it can be provided as a solution for navigation in open dynamic environment.

Despite the reasonable simulation results, our scheme has some limitations to improve and there still are more work to do.

5.1 Limitations

The "hesitation". As shown in our simulation results, when the robot and moving objects encounter almost synchronically, the robot will hesitate to navigate through, even though there might have chances for it to get navigable paths.

The constant velocity hypothesis. Our simulations assume the moving objects moves at constant velocity, which is reasonable in the open space when the moving objects are not intelligent. However, moving objects might have adaptive control scheme in some complex environment, and adjust its velocity according to environment, thus the hypothesis becomes futile.

5.2 Future Work

As shown in the last subsection, our scheme still has some limitations, there are more works to deal with those limitations.

Dealing with "hesitation". The hesitation appears because the path planning algorithm of robot always wants to get the optimal solution, which depends on predicted encounter time. In turn, the encounter time is computed by both the robot's and moving object's states. When the robot change its state according to previous path plan, the encounter time would be computed again, thus the path planning will

change meanwhile. Therefore, there is may have a potential solution with a non-optimal path planning.

Intelligent Objects. When the objects have the ability to change its motion according to its environment, we must take its reaction into consideration. In most cases, there are many choices for the robot and intelligent objects respectively, thus the situation could be modeled as a game, each has its various strategies, thus the game theory could be included in future attempts.

ACKNOWLEDGMENT

The authors would thank D. V. Lu， who has provided the open source of layer costmap code at website: https://github.com/DLu/navigation_layers. .

REFERENCES

[1] K. O. Arras, O. M. Mozos, and W. Burgard. Using boosted features for the detection of people in 2d range data. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3402–3407, April 2007.

[2] A. Elnagar and K. Gupta. Motion prediction of moving objects based on autoregressive model. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 28(6):803–810, Nov 1998.

[3] Ashraf Elnagar. A robust incremental algorithm for predicting the motion of rigid body in a time-varying environment. *International Journal of Intelligence Science*, 2(03):49, 2012.

[4] Ashraf Elnagar et al. Prediction of future configurations of a moving target in a time-varying environment using an autoregressive model. *Intelligent Control and Automation*, 2(04):284, 2011.

[5] Brian P Gerkey and Motilal Agrawal. Break on through: Tunnel-based exploration to learn about outdoor terrain. In *ICRA Workshop on Path Planning on Costmaps*, 2008.

[6] Chandimal Jayawardena. A technical review of motion prediction methods for indoor robot navigation. 2015.

[7] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A hybrid prediction model for moving objects. In *2008 IEEE 24th International Conference on Data Engineering*, pages 70–79, April 2008.

[8] Atif Khan, Febin Moideen, Juan Lopez, Wai L. Khoo, and Zhigang Zhu. *KinDectect: Kinect Detecting Objects*, pages 588–595. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[9] Sujeong Kim, Stephen J. Guy, Wenxi Liu, Rynson W. H. Lau, Ming C. Lin, and Dinesh Manocha. *Predicting Pedestrian Trajectories Using Velocity-Space Reasoning*, pages 609–623. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[10] R. Kirby, R. Simmons, and J. Forlizzi. Companion: A constraint-optimizing method for person-acceptable navigation. In *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pages 607–612, Sept 2009.

[11] D. V. Lu, D. Hershberger, and W. D. Smart. Layered costmaps for context-sensitive navigation. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 709–715, Sept 2014.

[12] L. Matthies and A. Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 727–733 vol.2, Apr 1988.

[13] Daniel Morris, Brian Colonna, and Paul Haley. Ladar-based mover detection from moving vehicles. In *Proceedings of the 25th Army Science Conference*, 2006.

[14] Luis E. Navarro-Serment, Christoph Mertz, and Martial Hebert. Pedestrian detection and tracking using three-dimensional ladar data. *The International Journal of Robotics Research*, 2010.

[15] P. Payeur, Hoang Le-Huy, and C. M. Gosselin. Trajectory prediction for moving objects using artificial neural networks. *IEEE Transactions on Industrial Electronics*, 42(2):147–158, Apr 1995.

[16] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B Cremers. People tracking with mobile robots using sample-based joint probabilistic data association filters. *The International Journal of Robotics Research*, 22(2):99–116, 2003.

[17] L. Spinello and K. O. Arras. People detection in rgb-d data. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3838–3843, Sept 2011.

[18] Yufei Tao, Christos Faloutsos, Dimitris Papadias, and Bin Liu. Prediction and indexing of moving objects with unknown motion patterns. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pages 611–622, New York, NY, USA, 2004. ACM.

[19] L. Xia, C. C. Chen, and J. K. Aggarwal. Human detection using depth information by kinect. In *CVPR 2011 WORKSHOPS*, pages 15–22, June 2011.